Technical Module Description: LEDControl.py

Version: 1.0 Date: January 29, 2025

Authors: Henk Stevens, Olaf Mastenbroek, Onno Janssen

Organization: Stichting Oradio

License: GNU GPL

# Module Purpose

This module (`LEDControl.py`) provides GPIO-based control for the LEDs on the Oradio device. It allows turning LEDs on or off, blinking them with configurable timing, and performing delayed LED control actions. The module includes safety handling to avoid conflicts between manual control and blinking operations. It is designed for use on Raspberry Pi platforms and integrates Oradio’s custom logging system via `oradio\_log`.

# Main Features

* LED On/Off control for individual LEDs.
* Blinking functionality with adjustable cycle time for each LED.
* Delayed LED turn-off after a defined active period.
* Prevention of blinking conflicts by managing separate threads per LED.
* Safe shutdown and GPIO cleanup on program exit.
* Standalone interactive testing via command-line menu for development and diagnostics.

# Key Components and Flow

## Initialization (\_\_init\_\_)

Sets up Raspberry Pi GPIO in BCM mode. Configures all LEDs as output pins and ensures they are initially turned off (GPIO HIGH). Prepares thread tracking dictionaries for managing blinking operations.

## turn\_on\_led(led\_name)

Turns on the specified LED and stops any active blinking thread for that LED.

## turn\_off\_led(led\_name)

Turns off the specified LED and stops its blinking thread if active.

## turn\_on\_led\_with\_delay(led\_name, delay)

Turns on the specified LED and automatically turns it off after a given delay (in seconds), using a separate thread.

## control\_blinking\_led(led\_name, cycle\_time)

Starts or stops blinking for a specified LED. Blinking is handled in a dedicated thread per LED, with the blink interval set by `cycle\_time`. Stops blinking if `cycle\_time` is zero or None.

## turn\_off\_all\_leds()

Turns off all LEDs and stops all active blinking threads.

## cleanup()

Stops all blinking, turns off LEDs, and performs GPIO cleanup to safely release GPIO resources.

# Usage Example (Standalone Test Mode)

When executed directly (`python3 LEDControl.py`), the module provides an interactive command-line menu to manually test LED functions. Users can select individual LEDs, turn them on or off, configure blinking, or set delayed turn-off behavior. This interactive mode is useful for development and hardware verification.